A BAT-INSPIRED T-WAY STRATEGY FOR MIXED-STRENGTH TEST SUITE GENERATION

YAZAN AHMAD AL SARIERA

Doctor of Philosophy

UNIVERSITI MALAYSIA PAHANG

SUPERVISOR'S DECLARATION

We hereby declare that we have checked this thesis and in our opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Doctor of Philosophy.

_____

(Supervisor's Signature)

Full Name : Prof. Dr. Kamal Z. Zamli

Position : Dean, Faculty of Computer Systems and Software Engineering, (Professor)

Date : 27 October 2017

_____

(Co-supervisor's Signature)

Full Name : Assoc. Prof. Mazlina Abdul Majid

Position : Deputy Dean, Faculty of Computer Systems and Software Engineering, (Associate Professor)

Date : 27 October 2017

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

_____

(Student's Signature)

Full Name : Yazan Ahmad Sadeq Al Sariera

ID Number : PCS14001

Date : 27 October 2017

A BAT-INSPIRED T-WAY STRATEGY FOR MIXED-STRENGTH TEST SUITE GENERATION

YAZAN AHMAD SADEQ AL SARIERA

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Doctor of Philosophy

Faculty of Computer Systems and Software Engineering

UNIVERSITI MALAYSIA PAHANG

OCTOBER 2017

DEDICATION

Dedicated to my parents.

For their endless love, support and encouragement.

ACKNOWLEDGEMENTS

First and foremost, I owe my deepest gratitude to Allah (SWT) for giving me live, strength and guidance throughout my study period and the ability to complete this thesis.

I am deeply grateful to my parents: my mother Eman Alsariera and my father Ahmad Alsariera for your prayers and unlimited support. I certainly would not be where I am today without their nurture, guidance, love and care throughout my life. There are not enough words in the world to express my appreciation. Whatever I do, I will not be able to return the love, bestowal and support. I pray Allah for your perfect health and prolong age and may Allah reward you richly. Love you both.

I am greatly indebted and appreciate very much to my brothers and sister for their encouragements, supports and sacrifices all the time. To my relatives and dearest beloved, a grateful thank you for your support and encouragement, may Allah bless you all.

I would like to express my gratitude to my research supervisor Prof. Dr. Kamal Zuhairi Zamli for his excellent ideas, profound technical vision, invaluable guidance, patience, motivation, many fruitful discussions and constant support in making this thesis possible. I am so much grateful for accepting me to become a part of his research team.

I would like to thank my co-supervisor Assoc. Prof. Mazlina A. Majid and Dr. AbdulRahman A. AlSewari for their support and the faculty members and staff for their hard work as well.

Finally, I would like to acknowledge and thank Universiti Malaysia Pahang for allowing me to conduct my research and providing the required assistance. In addition, I would like to thank the Ministry of Science, Technology and Innovation (MOSTI), Malaysia for funding this work.

TABLE OF CONTENT

DECLARATION

TITLE PAGE

ACKNOWLEDGEMENTS **iii**

ABSTRAK **iv**

ABSTRACT **v**

TABLE OF CONTENT **vi**

LIST OF TABLES **ix**

LIST OF FIGURES **x**

LIST OF ABBREVIATIONS **xii**

CHAPTER 1 INTRODUCTION **1**

1.1 Overview of Software Testing 1

1.2 Research Motivation 3

1.3 Problem Statements 5

1.4 Research Aim and Objectives 7

1.5 Research Scope 7

1.6 Operational Framework 8

1.7 Thesis Organization 9

CHAPTER 2 LITERATURE REVIEW **11**

2.1 The Test Case Design Strategies 11

2.1.1 Random Testing 11

2.1.2 Equivalence Class Partitioning 12

2.1.3 Boundary Value Analysis 13
2.1.4 Cause and Effect Graphing (CEG) 14
2.1.5 Interaction Sampling 16
2.2 The Mathematical Notations for *t*-way Test Suite Generation 16
2.3 A Problem Definition Model for *t*-way Test Suite Generation 19
2.4 Formal Definition for *t*-way 30
2.5 The Existing *t*-way Strategies 33
2.5.1 Deterministic *t*-way Test Suite Generation Strategies 35
2.5.2 Probabilistic *t*-way Test Suite Generation Strategies 37
2.5.3 The Observation of the Highlighted *t*-way Strategies 45
2.5.4 The Justification of the Adoption of BA 47
2.6 Summary 50

CHAPTER 3 RESEARCH METHODOLOGY 51

3.1 The Original BA Algorithm 51
3.2 The BTS Strategy 55
3.2.1 Input Analysis 57
3.2.2 Interaction Generation 60
3.2.3 Test Suite Generation 68
3.2.4 Tuning of BTS Variables 77
3.3 Prototype Implementation 84
3.4 Summary 85

CHAPTER 4 RESULTS AND DISCUSSION 86

4.1 Experimental Evaluations 86
4.2 Experimental Results 89

4.2.1 Characterizing BTS 89

4.2.2 Benchmarking with Other Strategies 91

4.2.3 Benchmarking for Mixed-Strength Test Configurations 97

4.3 Statistical Analysis of the Experimental Results 102

4.3.1 Statistical Analysis for *t*-way Results 102

4.3.2 Statistical Analysis of Mixed-Strength Results 106

4.4 Experimental Observation and Discussion 111

4.4.1 Experimental Results and Statistical Analysis Observations 111

4.4.2 Discussion 118

4.5 Summary 121

CHAPTER 5 CONCLUSION AND FUTURE WORK 122

5.1 Objectives Revisited 122

5.2 Contribution 124

5.3 Future work 125

REFERENCES 126

APPENDIX A THE RUNNING COMMAND-LINE FOR BTS 140

APPENDIX B BTS TUNING DATA 142

APPENDIX C THE LIST OF PUBLICATIONS AND AWARDS 161

LIST OF TABLES

Table 2.1	The grading and their corresponding letters.	13
Table 2.2	The decision table for the CEG in Figure 2-2.	16
Table 2.3	The analysis of existing t-way strategies.	47
Table 4.1	The characteristic of BTS (Hamming BA against original BA).	89
Table 4.2	The minimum test suite sizes for experimental set 1.	92
Table 4.3	The minimum test suite sizes for experimental set 2.	94
Table 4.4	The minimum test suite sizes for experimental set 3.	95
Table 4.5	The minimum test suite sizes for experimental set 4.	96
Table 4.6	The minimum test suite sizes for experimental set 5.	98
Table 4.7	The minimum test suite sizes for experimental set 6.	99
Table 4.8	The minimum test suite sizes for experimental set 7.	100
Table 4.9	The minimum test suite sizes for experimental set 8.	101
Table 4.10	Friedman test for Table 5.2.	103
Table 4.11	Wilcoxon signed-rank (Post-hoc) test for Table 5.2.	103
Table 4.12	Friedman test for Table 5.3.	104
Table 4.13	Wilcoxon signed-rank (Post-hoc) test for Table 5.3.	104
Table 4.14	Friedman test for Table 5.4.	105
Table 4.15	Wilcoxon signed-rank (Post-hoc) test for Table 5.4.	105
Table 4.16	Friedman test for Table 5.5.	106
Table 4.17	Wilcoxon signed-rank (Post-hoc) test for Table 5.5.	106
Table 4.18	Friedman test for Table 5.6.	107
Table 4.19	Wilcoxon signed-rank (Post-hoc) test for Table 5.6.	107
Table 4.20	Friedman test for Table 5.7.	108
Table 4.21	Wilcoxon signed-rank (Post-hoc) test for Table 5.7.	108
Table 4.22	Friedman test for Table 5.8.	109
Table 4.23	Wilcoxon signed-rank (Post-hoc) test for Table 5.8.	109
Table 4.24	Friedman test for Table 5.9.	110
Table 4.25	The experimental sets observation.	119
Table 4.26	The statistical significant achieved for each experimental sets.	120
Table A.1	The command-line specifications for BTS.	140
Table A.2	Examples of command-line specifications for BTS.	140
Table B.1	Full details of BTS tunning sizes and their averages.	142

LIST OF FIGURES

Figure 1.1	Overview of the software testing life cycle.	2
Figure 1.2	The CNC software system.	3
Figure 1.3	The parameters and their values for CNC software system.	4
Figure 1.4	The illustration of the operational framework for BTS.	8
Figure 2.1	A grade converter software.	12
Figure 2.2	The CEG for the Example in Figure 2.1.	15
Figure 2.3	The illustration of the mathematical notations.	18
Figure 2.4	Running example.	19
Figure 2.5	The running example parameters and values.	20
Figure 2.6	The exhaustive test suite (at t = 4) for CA(36,4, 2^2 3^2).	20
Figure 2.7	The illustration of the interaction elements set in (A) and the test cases set in (B) for 2-way interaction strength (at t = 2).	22
Figure 2.8	The running example interaction elements and test cases sets including randomized values.	23
Figure 2.9	Merging of all 2-way test sets, Final test suite for CA(11,4, 2^2 3^2).	24
Figure 2.10	Analysis of 2- way interaction configurations occurrence.	25
Figure 2.11	The mixed-strength interaction demonstration.	26
Figure 2.12	The 3-way interaction test suite for the first three parameters.	27
Figure 2.13	The test suite for the overall system with the mixed-strength interaction.	28
Figure 2.14	The element t-tuple sets dominastration.	31
Figure 2.15	The combinations t-tuples set (CTS) list illustration.	31
Figure 2.16	The illustration for IE set for EC 1 tuple in IET.	32
Figure 2.17	The illustration of the deterministic and probabilistic process.	35
Figure 2.18	Features of the existed *t*-way test suite generation strategies.	46
Figure 3.1	The BA pseudo code.	52
Figure 3.2	The overview of BTS strategy.	56
Figure 3.3	The illustration of the elements set based on the number of values for each elements.	57
Figure 3.4	The illustration of the variables processed in the input analysis phase.	58
Figure 3.5	The illustration of the mixed-strength configrations	58
Figure 3.6	Pseudo code of input analyser and legal value representation algorithm.	59
Figure 3.7	The illustration of EC matching and EC in CTS.	61

Figure 3.8	The pseudo code of CTS generator.	62
Figure 3.9	The construction of interaction elements and binary elements.	64
Figure 3.10	The pseudo code of IET generator that includes BES generation method.	65
Figure 3.11	The illustration of the CTS, IET and BES generation flow.	66
Figure 3.12	The pseudo code of test suite generation.	69
Figure 3.13	The illustration of test candidates mapping into the BA population.	71
Figure 3.14	The illustration of Hamming distance classifier.	74
Figure 3.15	The illustration of test suite generation mechanism.	76
Figure 3.16	The illustration of minimum sizes with 10 bats.	79
Figure 3.17	The illustration of sizes average with 10 bats.	79
Figure 3.18	The illustration of minimum sizes with 20 bats.	80
Figure 3.19	The illustration of sizes average with 20 bats.	80
Figure 3.20	The illustration of minimum sizes with 50 bats.	81
Figure 3.21	The illustration of sizes average with 50 bats.	81
Figure 3.22	The illustration of minimum sizes with 100 bats.	82
Figure 3.23	The illustration of sizes average with 100 bats.	82
Figure 3.24	The BTS prototype.	84
Figure 3.25	The functional hierarchy of BTS.	85
Figure 4.1	The convergence pattern.	90
Figure 4.2	The illustration of Table 5.2 results' intervals with CL 95%.	103
Figure 4.3	The illustration of Table 5.3 results' intervals with CL 95%.	104
Figure 4.4	The illustration of Table 5.4 results' intervals with CL 95%.	105
Figure 4.5	The illustration of Table 5.5 results' intervals with CL 95%.	106
Figure 4.6	The illustration of Table 5.6 results' intervals with CL 95%.	107
Figure 4.7	The illustration of Table 5.7 results' intervals with CL 95%.	108
Figure 4.8	The illustration of Table 5.8 results' intervals with CL 95%.	109
Figure 4.9	The illustration of Table 5.9 results' intervals with CL 95%.	110
Figure A.1	The BTS advance user prototype.	141

LIST OF ABBREVIATIONS

A_0	Loudness
A_i	Initial loudness
ABC	Artificial Bee Colony
ABC- CAG	Artificial Bee Colony-Covering Array Generator
ACA	Ant Colony Algorithm
ACA-Shiba	Ant Colony Algorithm implemented by Shiba
ACO	Ant Colony Optimization
ACS	Ant Colony System
ACS-Chen	Ant Colony Algorithm implemented by Bryce
ACS-VSITs	Ant Colony System Variable Strength Interaction Test suites
ACTS	Advanced Combinatorial Testing Suite
AETG	Automatic Efficient Test Generator
ANNs	Artificial Neural Networks
AR	Anti-random
BA	Bat-inspired Algorithm
BA*	Bees Algorithm
BC	Base Choice
BCAETG	Compound BC and AETG strategy
BE	Binary Element
BES	Binary Element Set
BKM	Bat-K-Means
BTS	Bat-inspired Testing Strategy
CA	Covering Array
CASA	Simulated Annealing Algorithm for constrained Combinatorial interaction testing
CATS	TestCover
CEG	Cause and effect graphing
CI	Confidence interval
CPU	Central Processing Unit
CS	Cuckoo Search
CTE-XL	Classification-Tree Editor eXtended Logics
CTM	Classification-Tree Method
CTS	Combination t-Tuple Set
CTS	Combinations t-Tuples Set
CTS*	Combinatorial Test Services
CNC	Computer Numerical control
DF	Degree Of Freedom
DPSO	Discrete Particle Swarm Optimization
E	Element
EC	Element Combination
EC's	Element Combinations
EGA	Evolutionary Genetic Algorithm
ES	Element Set
FPA	Flower Pollination Algorithm

FSAPSO	Fuzzy Logic Agorithm Particle Swarm Optimization
FTS	Final Test Suite
GA	Genetic Algorithm
GA-Huang	Genetic Algorithm strategy implemented by Huang
GAPTS	Genetic Algorithm for Pairwise Test Sets
GUI	Graphical User Interface
H_0	Null hypothesis
H_1	Alternative hypothesis
HC	Hill Climbing
HC-Bryce	Hill Climbing strategy implemented by Bryce
HHH	High Level Hyper-Heuristic
HS	Harmony Search
HS-PTSGT	Harmony Search-Pairwise Test Suite Generator Tool
HSS	Harmony Search Strategy
HSTSG	Harmony Search Test Suite Generator
IE	Interaction Element
IE's	Interaction Elements
IET	Interaction Element Tuples
HIS	Improved HS
IPO	In Parameter Order
IPOG	In-Parameter-Order-General
IPOG-D	In-Parameter-Order-General Double
ISA	Improved SA
ITCH	Intelligent Test Case Handler
JRE	Java Running Environments
LAHC	Late Acceptance Hill Climbing
LOC	Lines of Code
mAETG	Modified Automatic Efficient Test Generator
MCA	Mixed-level Covering Array
mMCA	Mixed-strength Mixed-level Covering Array
MIPOG	Modified IPOG
MOCell	A Cellular genetic algorithm for MultiObjective optimization
MOL	Many Optimization Liaisons
mCA	Mixed-strength Covering Array
mTCG	Modified Test Case Generator
NP	Non-deterministic Polynomial-time
NSGA-II	Nondominated Sorting Genetic Algorithm II
OA	Orthogonal Arrays
OATS	Orthogonal Array Based Testing Strategy
P	Parameter
PGAS	A Parallel Genetic Algorithm based on Spark
PHSS	Pairwise Harmony Search Strategy
PICT	Pairwise Independent Combinatorial Testing
PITS	Prioritized pairwise Interaction Test Suite
PPSTG	Pairwise Particle Swarm Test Generator
PPW	Partly Pair-Wise

PSO	Particle Swarm Optimization
PSO-Chen	Particle Swarm Optimization strategy implemented by Chen
PSTG	Particle Swarm Test Generator
PTSG-GA	Pairwise test set generator using genetic algorithm
Q	Frequency
r	Emission of pulse rate
RAM	Random Access Memory
RTS	Reverse Tracking Strategy
SA	Simulated Annealing
SA-Bryce	Simulated Annealing strategy implemented by Bryce
SA-Mayer	Simulated Annealing strategy implemented by Mayer
SAVNS	Simulated Annealing Variable Neighbourhood Search
SBC	Simulated Bee Colony
SBSE	Search-Based Software Engineering
SSO	Simplified Swarm Optimization
T	Interaction strength
tCA	Improved CASA
TCG	Test Case Generator
TConfig	Test Configuration
T_{max}	Number of generation (iteration)
TS	Tabu search
ts	Time Step
TS-Bryce	Tabu Search strategy implemented by Bryce
TVG	Test Vector Generator
v	Value
MC	Mixed-strength condition
VSITs	Variable Strength Interaction Test suites
VS-PSTG	Variable Strength Particle Swarm Test Generator
X	Position (location)
X^2	Chi-square
a	Alpha